

Hierarchical Fuzzy Deep Learning for Image Classification

Shashank Kamthan¹, Harpreet Singh², Thomas Meitzler³

^{1,2}Wayne State University, Detroit, MI, USA 48202; ³Ground Vehicle Survivability and Protection,
CCDC Ground Vehicle Systems Center, Warren, MI, USA

ABSTRACT

Considerable interest has been shown for the last several decades for fuzzy logic and its application. The intelligent systems and deep learning systems are gaining breakthroughs in all walks of life to solve the real-life problems for future. The conventional fuzzy has the constraint to work with limited rule dimensions, whereas deep neural networks are unable to handle uncertain and imprecise data implicitly in the system. The main objective of this paper is to develop a generalized algorithm for intelligent systems that can handle uncertainty and imprecise behavior especially for processing of large image datasets. In this paper, the hierarchical fuzzy approach is suggested, as it is gaining attention to tackle large real-life problems. The strategy used is to partition large image dataset into small data samples and connect all the fuzzy subsystems in a hierarchical manner. To the best of authors' knowledge, nobody has developed a hierarchical fuzzy approach to handle large image dataset of real images. The algorithm for hierarchical fuzzy logic for large image data using image thresholding has been discussed. To make the assessment, the real image database has been considered. The image classification has attained the potential applications to defense and security especially for the target identification and classification. The accuracy and computational time comparisons of hierarchical fuzzy systems with existing methodologies such as deep neural networks have been discussed.

Index Terms—fuzzy systems, hierarchical fuzzy, hierarchical deep learning, big data, image classification, image identification

I. INTRODUCTION

“Fuzzy Logic” term was first introduced in 1965 by Lotfi Zadeh [12] to handle uncertain, imprecise or vague information. Fuzzy comprises the use of linguistic parameters that bridge the gap between mathematical and realistic systems. Fuzzy logic originates from the generalization of conventional logic. It runs by the smooth transition between true and false. The conventional logic presents output as ‘0’ or ‘1’, whereas fuzzy logic presents the result as degree of truth in range [0, 1]. Fuzzy logic applies the theory of partial truth, where output may oscillate between complete true and complete false.

The fuzzy system is designed in three steps: fuzzification, inference and defuzzification. Fuzzification converts the input into suitable linguistic values that can be viewed as labels for fuzzy sets. Inference systems possess a rule base containing the selection of input-output relations. Data set for the rules comprises membership functions and normalization of input and output combination. Defuzzification combines the output from the inference system and converts it into non-linguistic terms.

With the passage of time, several offshoots of fuzzy systems have been developed, such as, neuro-fuzzy systems or fuzzy neural networks, fuzzy clustering, fuzzy logic image processing etc. With the coming of neural networks, the concept of fuzzy logic has been extended to neuro-fuzzy logic. Fuzzy logic provides effective resolution and better assessment for the computational problems that includes uncertainty, imprecision and vagueness of human assessments. On the contrary, other approaches such as neural networks have limitations to handle fixed and precise data only.

Similar to the fuzzy systems, the hybrid approaches such as FNN (Fuzzy neural network) or NFS (Neuro Fuzzy systems) integrate neural networks and fuzzy logic behavior. This hybrid approach adds linguistic reasoning from fuzzy logic and cognitive connections from neural networks. The hybridization results in an intelligent system. These systems embrace fuzzy set theory in

which the system contains various rules with the linguistic set of information defined in IF - then - ELSE format. The design of neural-fuzzy systems focusses on interpretability and accuracy. This is why the neuro-fuzzy systems have been more favored than fuzzy systems.

Major limitation of fuzzy systems is data handling where the fuzzy logic limits the size of data in the system. With proven techniques, neuro-fuzzy has the ability to handle computation with a large number of rules. However, there has been a challenge in using neuro-fuzzy systems to solve complex and high input-output dimensions issues because of working limitations to certain dimensions. Recently, several authors [8] [12] have worked on and are still working on developing deep neural networks techniques to handle the problem of classification of images for intelligent systems. However, neural networks do not drive imprecise and uncertain data, which is prevalent in a large number of real-life conditions. In order to handle any imprecision and uncertainty, neuro-fuzzy logic techniques have been promoted.

Neuro-Fuzzy logic adheres to white-box system. These systems bridge interpretation between linguistic form and mathematical analysis and also provide transparency. These systems have been commonly used to design complex systems but worked only within finite rule and data dimensions. These limitations have become a bottleneck in several highly complicated real-life scenarios. These limitations reduce generalization and transparency of the overall system. The major factors affecting limitations of input-output dimensions for a fuzzy and neuro-fuzzy system can be briefly defined as the exponential increase in rules with increase in input parameters, datasets or input-output combinations and mathematical relations.

To overcome the issues of generalization, transparency and dimensions, various researchers ignite the idea of incorporating hierarchical systems with fuzzy logic. Instead of designing conventional systems with high dimensions, the system will be divided into subsystems with low dimensions and interconnect with each other in the form of hierarchy structure. The hierarchical system is represented as a multi-input single-output system. Without losing generalization, multi-input-multi-output systems are represented in a number of multi-input-single-output subsystems. The common applications to hierarchical systems are classification, clustering, planning and tracking systems etc. Due to the limitation of fuzzy logic rule-base for large dataset, this paper is presenting an approach to design hierarchical fuzzy systems using image thresholding that will handle large image dataset. This approach will reduce the number of rules in total rule-base without compromising effectiveness and efficiency of the system.

The complexity of systems has been best represented by a total number of rules. Increases in the number of inputs lead to increase in total rules, thus increasing the overall complexity of the system. This limitation restricts the usage of fuzzy systems to solve complex problems and real-life applications with large dimensions especially with large images. It thus increases the number of rules because the total number of rules are directly proportional to the increase in number of input parameters and complexity of the system.

In the last few decades, the hierarchical fuzzy system has appeared to be a viable solution to overcome the limitation of conventional fuzzy and neuro-fuzzy systems. To the best of the authors' knowledge, nobody has exploited hierarchical fuzzy logic for large image datasets by partitioning the dataset into small data samples and synthesizing it. It has been demonstrated by several authors [1] that the hierarchical system has reduced the overall complexity of the system as it results in a smaller number of rules compared to conventional fuzzy logic. Besides complexity, the hierarchical system reduces computation time by calibrating the number of rules required for each fuzzy logic unit in a hierarchical structure. With these factors in mind, we tend to use hierarchical fuzzy for the use in large image dataset problems.

Fuzzy logic has been applied to number of real-life applications by various researchers [5] in the field of engineering, technology, industry, science, mathematics, business analytics, data analytics, defense and security, medicine and biomedical fields, internet and computer security, power industry, consumer electronics, weather forecasting, business etc. One of the important applications of fuzzy in defense is the survivability of manned, robotic, and optionally manned vehicles. This paper presents an approach to develop an algorithm with the help of hierarchical fuzzy systems.

Section II describes various hierarchical tree structures along with mathematical representation and discusses various real-life applications of hierarchical fuzzy systems. Section III presents the systems design of hierarchical fuzzy logic. The steps to design an algorithm using image thresholding and conceptual flowchart have been discussed and steps to generate fuzzy logic systems using fuzzy c-mean clustering have been shown. Section IV shows the description of an image database such as YaleB database.

Section V shows the results of the generalized algorithm for hierarchical systems and its comparison with a conventional system such as neural networks etc.

Problem Statement

Despite several advancements discussed in literature, still there are several open points and limitations to deep learning techniques. Some of the existing techniques such as neural networks have a constraint of handling only fixed and precise data, and obstruction of data, which is uncertain, imprecise and vague in nature. These limitations reduce the effectiveness and performance of the system and thus minimize the ability of deep learning techniques to confront real-life applications, which can be safety and security critical. This requires the necessity to illustrate data in a better way by collection of additional data, more analysis and clean dataset. Deep learning approaches often have low transparency. This reduced transparency limits the understanding and interpretation to other users. To confront these limitations, fuzzy logic presents an effective solution and better assessment of human interpretations that includes uncertainty, imprecision and vagueness.

It is a well-accepted fact that the conventional fuzzy systems have limitations to the rule dimension. The limitation restricts the usage of fuzzy systems to solve complex problems and applications with large data dimensions. In the last few decades, hierarchical fuzzy systems have appeared to be a viable solution to overcome the limitations of conventional fuzzy systems and, so far, bring significant attention. This paper presents an approach to develop hierarchical fuzzy systems with the focus to deal with large volumes of data, especially large image datasets, without compromising the effectiveness and accuracy of the system. The multidimensional image thresholding has been considered to modify input parameters, and then segment image samples. With the ability to maintain composition both at functional level and linguistic level, hierarchical fuzzy systems are well suited and have potential to handle large dataset efficiently and provide a high transparency due to its efficient rule base nature. The recent development has been around hierarchical fuzzy systems or fuzzy hierarchy networks.

There is no such algorithm available in literature to support the design of both type-1 and type-2 fuzzy inference systems using hierarchical structure. In this paper, the algorithm presented for hierarchical approach has been studied in context of both type-1 and type-2 fuzzy inference systems. The type-1 and type-2 fuzzy inference system differ by membership function definitions and representation of rules. The type-1 fuzzy system has crisp membership definitions; however, the type-2 fuzzy inference system has fuzzy membership definitions in nature. The advantage of using type-2 fuzzy over type-1 fuzzy is its ability to manage uncertain and irregular environments. Type-2 fuzzy [9] has high computation due to a large number of parameters required to design and also it requires the type-reduction mechanism in its defuzzification level. The membership upper membership function. Type reducer defuzzification produces the output by averaging the intervals at various stages. In this paper, type-1 fuzzy has been considered but the same algorithm can be used for the type-2 fuzzy inference system.

Authors' Contribution

The subject of deep learning has attracted the attention of a large number of researchers in different fields. The neural networks are considered as the most common deep learning technique. The neural networks do not consider imprecise, uncertain and vague human behavior. For example, the neural networks for the image identification and classification possess a constraint to handle only fixed and precise data, whereas various complex real-life applications are uncertain, imprecise and vague in nature along with human behavior. Whereas, the fuzzy logic presents an effective solution and better assessment for the complex problems, which includes uncertainty, imprecision and vagueness of human assessments. Because of the limitation of neural networks, it is important to develop a new methodology such as fuzzy deep learning. The subject of fuzzy deep learning appears to be in infancy. To surpass the recent issues such as handling big data, imprecision and uncertainty including human behavior, it is of significant importance to develop the new deep fuzzy learning techniques. The subject of image processing has found applications in a variety of fields including defense and security. To develop algorithms for advanced big databases of image applications having imprecision and uncertainty, the exploitation of fuzzy deep learning for image processing is considered decisive. Keeping this in view, this effort has been made to develop algorithms for large image data applications, which consists of imprecision and uncertainty including human behavior. It has also been established that with the increase in sophistication of data, the rules and dimensions in fuzzy logic have gone beyond the limits. The hierarchical fuzzy logic is one such technique, which is helpful in reducing the number of rules dimensionality and complexity of the real-time image processing applications. In this paper, we have taken data from social

networks to validate the methodology. The data for social networks is available for research purposes in a number of applications, however the data with imprecision, uncertainty and human behavior is barely available. Because of this limitation, this paper takes into the account the database from YaleB for facial recognition, which is readily available for research purposes.

The main objective of this paper is to develop a generalized algorithm for large intelligent systems using hierarchical fuzzy logic, which can handle uncertainty and imprecise behavior of the system and explore its application to image processing. The main contribution in this paper is the systematic and methodical development of the algorithm to design a hierarchical fuzzy system of image identification and classification for large image datasets. Otsu's multidimensional approach is used to identify multi dimension threshold values. These values will then be utilized to generate modified input. The output labels remain the same. The approach presented in this paper is to partition large dataset into small data samples and generate fuzzy systems (or fuzzy units) for every small sample. Later, depending on hierarchical tree structure, these fuzzy logic units are connected in a desired hierarchical manner. This approach addresses and eliminates the limitations of rule dimension possessed by conventional fuzzy systems. The algorithm has been validated on real-time image big dataset such as YaleB database etc. The generalized algorithm can be implemented using any of the hierarchical tree structures. The accuracy and computational time have been compared between hierarchical systems and existing methodologies such as convolutional neural networks. It is clearly shown that the hierarchical approach presented in this paper has much better and reduced computational time compared to neural networks.

II. HIERARCHICAL FUZZY SYSTEMS

Hierarchical systems, as defined in wiki, is an arrangement of items (objects, names, values, categories, etc.) in which the items are represented as being "above", "below", or "at the same level as" one another.

There has always been a huge challenge to solve complex issues by fuzzy or neuro-fuzzy, if the dimension and size of the dataset is very large. One prominent way to resolve this issue is to divide the system into small subsystems and arrange in the form of specific structure, where every subsystem is defined by a fuzzy system. Such system arrangements are commonly known as hierarchical fuzzy systems [2] and appear to be the most viable and productive option.

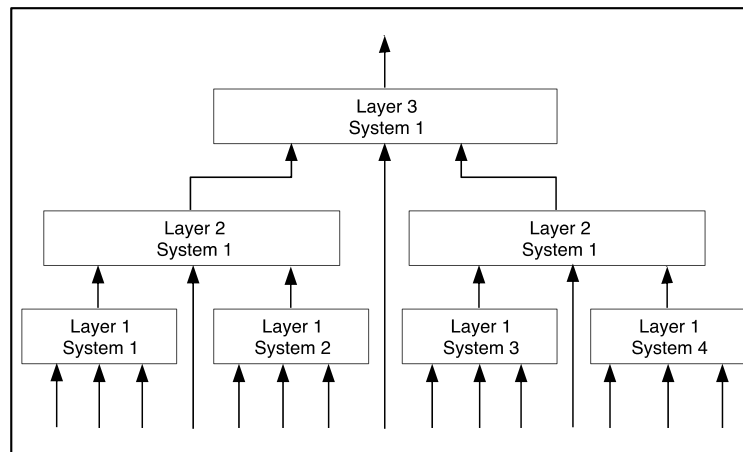


Fig 1. Cascaded structure of hierarchical systems

In the early 90's, to encounter the problems such as overflow of systems, losing generalization and transparency etc., Raju, Zhou and Kisner [2] presented the idea of hierarchical fuzzy systems [2] [3] [4]. As an alternative to design and develop conventional fuzzy systems with very high dimensions, a number of sub-categorized structures, commonly known as a fuzzy logic unit, with low dimensions are connected in the form of hierarchy. In some areas, various researchers consider hierarchical systems on top of fuzzy logic to refine final output. Common applications to hierarchical fuzzy systems are classification, clustering, planning and tracking systems etc. The system presented as a MISO i.e., multi-input and single output system, whereas multi-input and multi-output systems, without losing generalization, can be represented in a number of MISO subsystems. The lowest level of the hierarchical structure receives access to real inputs whereas all the consecutive levels are connected to output from previous level and real inputs both.

Depending on the architecture, the hierarchical fuzzy can be described in various levels. Every level comprises several fuzzy logic units. Three types of hierarchical structure approaches are commonly in use: incremental, aggregated and cascaded. The structures below are also referred to as multiple-stage structures where every level represents one stage. Assume, one fuzzy system defines every stage and the next level consumes the output from the previous level along with original input parameters. To maintain accuracy of the system, it is highly important to consider important inputs for the first stage and less important parameters shall be defined in later stages. Fig 1 shows the cascaded structure of hierarchical systems.

Incremental structure of the hierarchical system is first proposed by Chung and Duan [2] in the year 2000, as shown in Fig 2. While designing a system with incremental hierarchical structure, it is critical to rank the inputs based on their importance and effectiveness on overall system behavior.

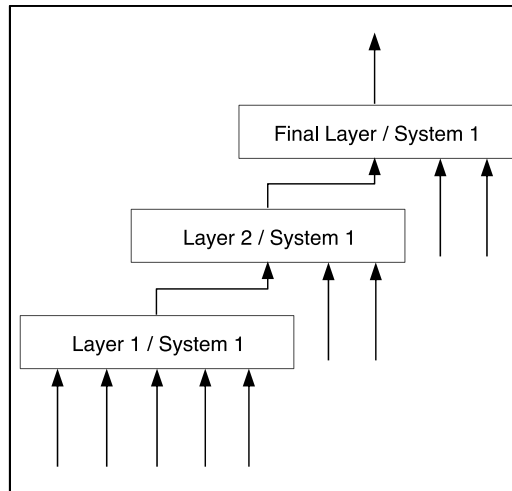


Fig 2. Incremental hierarchical structure

Wang [2] proposed another architecture commonly defined as aggregated hierarchical structures, shown in Fig 3. In this architecture, every stage can have multiple fuzzy systems, and most importantly all the original inputs are handled at the lowest stage. In the next stage onwards, outputs from previous systems are managed by next stage fuzzy systems. While designing a system with aggregated hierarchical structure, all the correlated input parameters are coupled in the same stage, whereas all the next levels depend completely on output of previous level as an input but with expected output to make the system accurate.

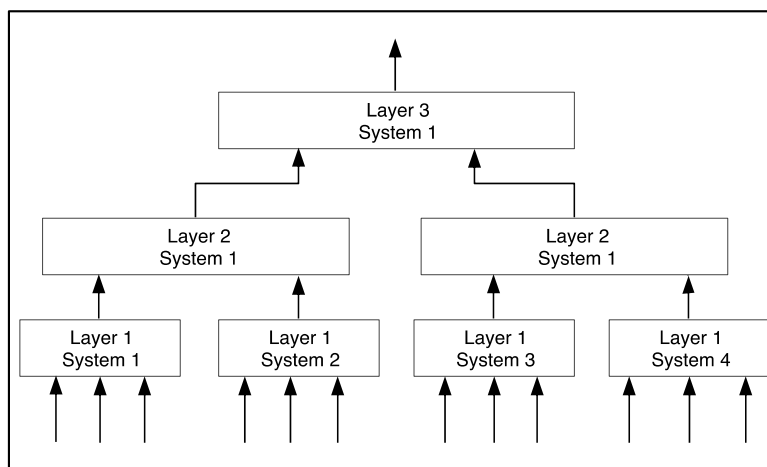


Fig 3. Aggregated hierarchical structure

It is noted that there is no direct method to identify the proper architecture of hierarchical structure or how many levels shall be considered, and also how many fuzzy systems may lead to required accuracy. These selections are made depending on the iteration and the system complexity / performance requirements. In literature review, no authors gave any methodical approach to design hierarchical systems for large dataset especially in case of images. In this paper, the authors present a systematic and generalized

DISTRIBUTION A. Approved for public release; distribution unlimited. OPSEC# 4809

approach for the development of hierarchical systems for big image datasets using image thresholding. Any of the given hierarchical structures can be designed and implemented by the given algorithm. In literature, mathematical representation of hierarchical fuzzy systems has been presented by various authors [1][17].

To understand better about the rule reduction using hierarchical fuzzy [1], refer to the 'Nxl' system, where 'N' is the total number of inputs and every individual input has defined by 'M' membership functions. The membership functions have been used to create input segments. To form a rule-base, these membership functions construct feasible interconnections between different inputs and outputs. Considering a conventional fuzzy system, the total number of rules possible can be represented as ' M^N '. However, for hierarchical fuzzy systems, the total number of rules possible can be represented as ' $(N-1).M^2$ '. Where, it is assuming that two input parameters have been used by every fuzzy logic unit of hierarchical systems. This leads to a maximum of '(N-1)' fuzzy units to cover 'N' number of input parameters. It is clearly observed that the total number of rules reduces by using hierarchical fuzzy systems compared to conventional systems because: $(N-1).M^2 \leq M^N$.

Applications of hierarchical systems

Most of the real-life applications and problems are very much evolved and have imprecise and uncertain behavior. Few examples are with the recent transformation in the vicinity of wireless sensor networks (WSNs), network transformation via Internet of things (IoT) etc. These applications possess large datasets that lead to drastic increase in rule dimensions and thus increase the complexity of the system. The conventional systems are unable to handle large dimension datasets. In the last few decades, the increase in complexity of real-life applications becomes uncontrollable and goes above the capabilities of conventional systems. Nevertheless, the hierarchical systems provide a relevant solution to these limitations possessed by conventional systems due to its capacity to reduce the complexity of systems, ability to handle large dimension datasets and uncertainty.

Few other real-life applications of hierarchical systems are in data mining, which necessitate multi-objective programming for both Type-1 and Type-2 fuzzy logic systems [9]. The hierarchical fuzzy systems have been efficiently used as multi-agent architectures for the parallel processing of search problems and distribution of task optimization between structure and set of parameter spaces.

Another application of the hierarchical fuzzy system is the design of adaptive Kalman filtering [14]. One of the major objectives of hierarchical fuzzy systems is to administer effective presentation irrespective of large dimension datasets. The systems consist of membership functions in reference to the concept of impreciseness, vagueness and uncertainty. By mathematically merging functions with other systems, complex abstractions can be achieved. The conventional system for Kalman filters is incapable to handle imprecise and uncertain behavior. However, hierarchical fuzzy systems have the ability to handle imprecise and uncertain behavior. Keeping similar considerations, the integration of the concept of both Kalman filter and hierarchical systems can be obtained for the estimation processes. This hierarchical fuzzy inductive inference can be beneficial for complimenting the probabilistic theory approach and it is more reliable and decisive while relating to non-linear relations.

III. ALGORITHM DESIGN USING HIERARCHICAL FUZZY LOGIC

For a small set of data, there are several algorithms available in the literature on designing hierarchical fuzzy systems. One of the algorithms was presented by Radek Sindelar [6] on hierarchical fuzzy systems. In this paper, the clustering approach [13] is used to design a fuzzy inference system for multiple subsystems and the center points of the cluster represent connection or rule base. The limitation of the algorithm is restricted to a small number of input parameters and to apply the same approach for large dataset is not a viable option. The paper also contains the analysis of the various weighted approaches such as maximum, weighted average etc. The connection or link between subsystems is made in such a way that the output of one subsystem becomes the input to another subsystem.

A large dataset such as an image database possesses an immediate challenge to a conventional fuzzy logic and now requires a distributed strategy to handle the data. Many companies such as Facebook, Google, Yahoo, Amazon etc. already possess large amounts of dataset i.e., images for various objects etc. and started tapping benefits from these data. In general, the raw data is very robust and complex. These variations in data representations present a unique challenge in designing any stable system. Deep

learning algorithms for neural networks, neuro-fuzzy, neural etc. show meaningful abstraction representation of raw data via hierarchical multi-level approach. This paper presents a methodology to design hierarchical fuzzy systems for a large set of real images. Various clustering approaches have been suggested by several authors. In this paper, we use the approach of fuzzy c-mean clustering [13]. The algorithm starts with the lower level, where inputs are real images. Just as a conventional fuzzy system has various types of membership function, similarly hierarchical fuzzy systems can be implemented by the same types of membership functions. In this paper, the system has been implemented using 2-side Gaussian membership function because of its smooth, continuous, differentiable characteristics.

Similar to conventional fuzzy, each hierarchical fuzzy sub-system [3] [4] is described in three blocks as: Fuzzifier, Rule editor and Defuzzifier. As a weighted approach, the “centroid” method has been considered as a defuzzification method. The simulation of all the subsystem in form of equation (2) and (3) looks like:

$$S_i = R^n \rightarrow R^m \quad \dots\dots\dots (2)$$

$$Z_i = [A_1, A_2, A_3, \dots, A_m] \quad \dots\dots\dots (3)$$

Where, ‘i’ is the number of subsystems, m is the number of clusters represented by multidimensional clusters. S_i and A_i are the membership functions represented by clusters. The approach suggested a cluster approach to define subsystems, extracted from image data. The common steps for the realization of hierarchical fuzzy logic unit are defined as: derive rules for training network, tune neural network parameters using fuzzy logic, add fuzzy logic criteria to define network, realize fuzzy sub-category such as membership functions through various algorithms (Grid partitioning, Back propagation, Subtractive Clustering and Fuzzy c-mean clustering) and represent fuzzy blocks for multi-layer networks (fuzzification, inference systems including rules and connections and defuzzification).

A. Image segmentation

The segmentation of images is the process to partition a single image in multiple objects, commonly known as image segmentation [8] [11]. The basic usage of image segmentation is to streamline image representation for better analysis i.e., identify objects, detect boundaries etc. This process assigns a label to every pixel but with varied characteristics. The behavior can be extracted from different sets of contours obtained from Image. The basic structure of image segmentation is image thresholding, which turns gray-scale or RGB image into binary form. There are several methods available in the literature for image thresholding, few of them are listed below as:

1. *Entropy Method* - This method uses the entropy of both foreground and background regions using the probability distribution functions. Assuming the foreground has largest entropy compared to background or previous ground
2. *Balanced Histogram Thresholding* - This method uses Iteration thresholding method. The image is classified into two regions front- and back-ground and weighs on histogram to see which side is heavier and then to make it lighter, removes the weight from heavier side.
3. *K-Means Clustering* - This method is based on iterations. Every set of image-pixels is divided into K-clusters that refers to the minimum distance between pixel and cluster center point. In the final step, apply any weighted approach i.e., Centroid, Maximum, Minimum etc. to get the final result.
4. *Otsu’s Method (Maximum Variance)* - For Image processing, Otsu’s method commonly referred after Nobuyuki Otsu is used for Image thresholding. The algorithm provides intensity or threshold values that differentiates between two-pixel classes. This approach determines the intra-or inter- class intensity variance and classifies image thresholding points where the variance is high.

Because of the capability of differentiating image pixel classes, in this paper, Otsu’s multi-dimensional method has been used to create the generalized algorithm using hierarchical fuzzy logic for a large database of images. The one-dimension Otsu’s method has limitations to have assumption of bi-model distribution. This method applies better when there is a deep and sharp curve. The problems discussed in this paper can be resolved by a multi-dimensional method and thus the reason behind using the multi-dimensional method in the algorithm. It is observed that in various databases available for research, there is a minimal difference in membership values between two image samples. This restricts the segmentation of data matrix and contradicts the design of fuzzy logic system rule base. With similar membership values, actual outputs are the same, but the expected outputs are different. One of the solutions to overcome this problem is to use Otsu’s image thresholding algorithm to find multiple threshold values and

multiply these values with input membership values to create the new input matrix but keeping output membership values remain the same and is shown by equation (4) below. Where, ‘N’ = number of columns for input matrix per sample; and ‘P’ = multi-threshold values from single sample input matrix.

$$\sum_{n=1}^N X_n^{New} = \sum_{n=1}^N \prod_{p=1}^P T_p * X_n \quad \dots (4)$$

B. Algorithm design

To design the sub-system, any one of four clustering methods (i.e., grid partitioning, back propagation, subtractive clustering and fuzzy C-mean clustering) can be picked. In this paper, Fuzzy C-Mean clustering [12] is considered due to its ability to generate center points. These points can directly be adapted in a fuzzy rule base. With required cluster center points, and using fuzzy c-mean clustering method, these regions are contrived to n points, where ‘n’ is the number of rules required for a fuzzy logic unit. The detailed steps for the algorithm to design hierarchical subsystems [1] [2] [3] are given below:

STEP [1]. Consider training data consisting of a large database with images i.e., YaleB database etc.

STEP [2]. Define following information:

- Number of hierarchical levels
- Fuzzy logic units in every level
- Number of clusters corresponds to required number of rules for every fuzzy logic unit

STEP [3]. Two sub-steps are defined below:

- Obtain multi-thresholding value from Otsu’s method and multiply with each input elements to create new modified input element
- Group the input dataset in various segments keeping the output remaining the same as expected. For instance, the input dataset of [165, 1024] can be divided in to 1408 samples of [15, 8] or 704 samples of [15, 16]

STEP [4]. For the first or lower level of hierarchical structure i.e., aggregated tree structure, all the real input adheres to the first or the lower level of hierarchical structure. This layer consists of fuzzy logic units generated for all the input samples extracted in Step 3. The steps to generate fuzzy logic unit are as follows:

- Take input segment / sample and map with desired output
- Using MATLAB commands, the syntax to generate Fuzzy inference system is defined below:
 - Fuzzy System = genfis3([Input], [Output], ‘FIS Type’, ‘Number of Cluster’); Where –
 - [Input] = Input sample matrix
 - [Output] = Expected output matrix
 - ‘FIS Type’ = Choose type of inference system: Mamdani or Sugeno
 - “Number of Clusters” = Desired number of rule base for fuzzy logic unit
 - Evaluate the output of the fuzzy logic unit with ‘evalfis’ command
 - Store the evaluated output in a buffer

STEP [5]. For the next level(s) of hierarchical structure,

- Take all the output from previous hierarchy levels and consider them as input to current level
- Create a group of segments of matrix and map each segment with map with desired output
- Using MATLAB commands, the syntax to generate Fuzzy inference system is defined below:
 - Fuzzy System = genfis3([Input], [Output], ‘FIS Type’, ‘Number of Cluster’); Where –
 - [Input] = Input sample matrix
 - [Output] = Expected output matrix
 - ‘FIS Type’ = Choose type of inference system: Mamdani or Sugeno
 - “Number of Clusters” = Desired number of rule base for fuzzy logic unit
 - Evaluate the output of the fuzzy logic unit with ‘evalfis’ command
 - Store the evaluated output in a buffer

STEP [6]. For the last hierarchy level

- Take all the output from previous hierarchy levels and consider them as input to current level
- Create a group of segments of matrix and map each segment with map with desired output
- Using MATLAB commands, the syntax to generate Fuzzy inference system is defined below:

- Fuzzy System= genfis3([Input], [Output], 'FIS Type', 'Number of Cluster'); Where –
 - [Input] = Input sample matrix
 - [Output] = Expected output matrix
 - 'FIS Type' = Choose type of inference system: Mamdani or Sugeno
 - "Number of Clusters" = Desired number of rule base for fuzzy logic unit
- Evaluate the output of the fuzzy logic unit with 'evalfis' command

STEP [7]. The evaluated output at the last level of hierarchy provides final output.

The above seven steps describe the complete design of the hierarchical system. The step 1 comprises the consideration of training data using which fuzzy logic units shall be generated. The step 2 contains the given conditions of the overall hierarchical system such as number of levels required for hierarchical design, number of units per level, number of rules per fuzzy logic units etc. The step 3 addresses the solution to handle the limitation of membership values for images. In this step for every input sample, multidimensional threshold values are calculated using the Otsu's multidimensional method and then multiply these threshold values with every individual element of input same. This step provides a completely new range of inputs for the system. The step 4-6 describes the process of sampling the dataset into small sub-samples and then using fuzzy c-mean clustering, generating a fuzzy inference system. This fuzzy inference system represents the fuzzy logic unit for hierarchical structure. The step 7 summarizes the final output in reference with the highest level of a hierarchical structure. The pseudo code for the above-mentioned steps [1] is given as follows:

Pseudo Code for Hierarchical System

STEP [1]. Data: Training data of large set of real images

STEP [2]. Thresholding: Find the image thresholding parameter for each sample and multiply with every element of the same sample. The result becomes the modified input parameters, and the output remains the same.

Segmentation: Using Mathematical analysis, create and pool similar inputs and pair them with the final output

Hierarchy: Define number of hierarchy level required

Find Cluster: Find center points (relation between Input and Output) for each group

STEP [3]. Grouping: Divide the data in number of columns and rows but with same output

STEP [4]. –STEP [16].

Execution:

- Assume the type of membership function is Gaussian and every Input / Output has defined MFs
- For h = 1 to Number of Hierarchy
 - If h == 1
 - Given: Take real input and output data
 - Calculate multi-threshold points of every sample (i.e., every row = 1 sample)
 - Multiple every input element for every sample with its respective multi-threshold values
 - Assume these values as modified input and use this for further calculation
 - Generate fuzzy Inference systems for all the groups. The syntax is:
 - For I = 1 to Number of Groups**
 - Find center points (IO Mapping)
 - Generate Fuzzy inference system
 - Evaluate FIS using "evalfis"
 - End process**
 - Else
 - Given: Output from previous level becomes input to the next level, but the output is real
 - Generate fuzzy Inference systems for all the groups. The syntax is:
 - For I = 1 to Number of Groups**
 - Find center points (IO Mapping)
 - Generate Fuzzy inference system
 - Evaluate FIS using "evalfis"
 - End process**

STEP [7]. Result:

- Evaluate final output
- Comparison between actual testing labels and generated output

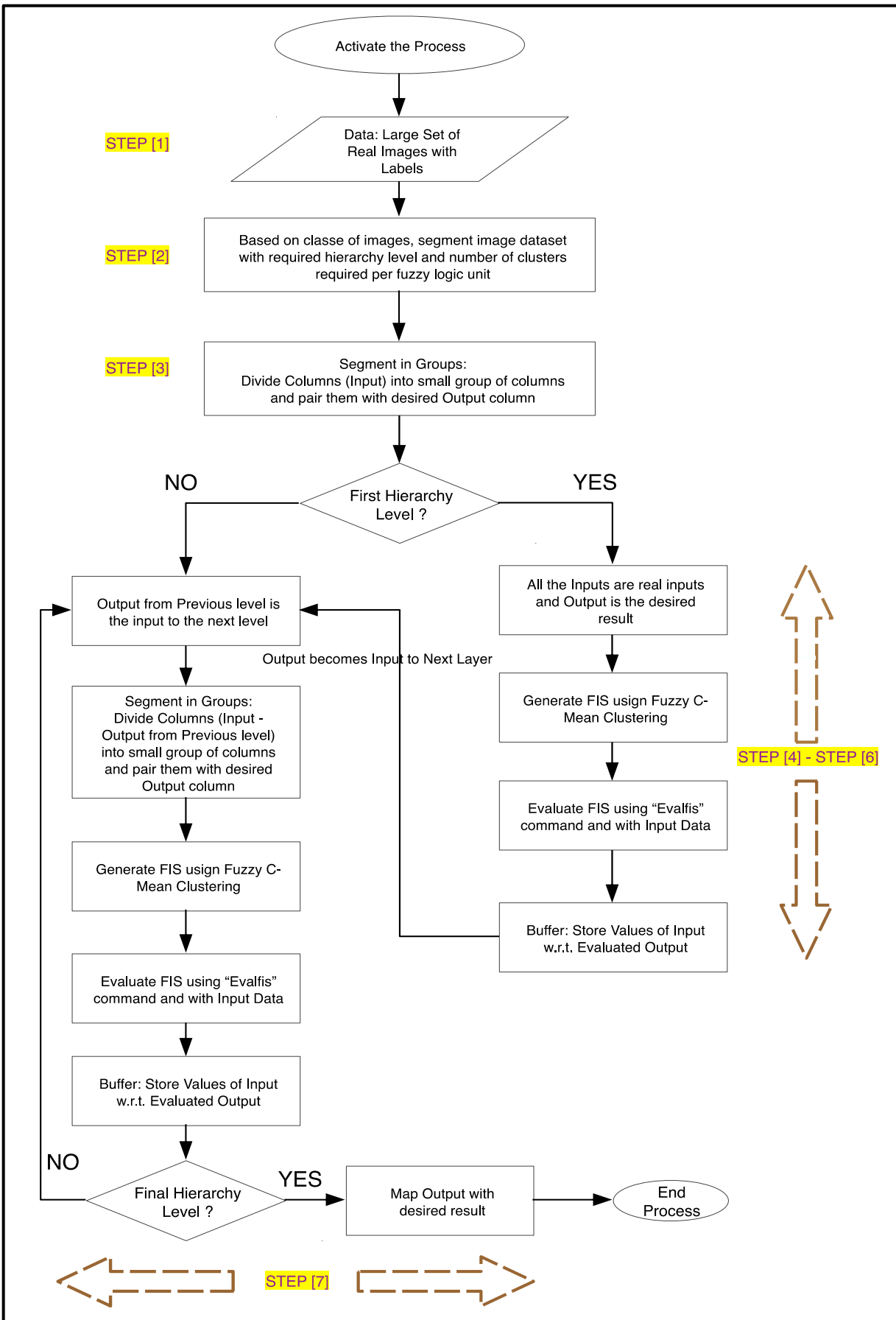


Fig 4. Conceptual hierarchical fuzzy logic implementation for the system

In this paper, aggregated hierarchical fuzzy tree structure has been considered since it provides results with more accuracy in reference to desired output. The lower level consumes the real images input data along with the expected output. The column matrix of input will then be segmented into various samples, keeping the same output label, and then design the fuzzy inference system (FIS) for each sub-system. The outputs of these fuzzy inference systems will be collected together in the form of matrix, as these outputs from the previous level will be input to the next level. From the second hierarchy level onwards, repeat the same process as below. For the last hierarchical level, there will be only one fuzzy logic unit that provides final output. This output will be mapped to the classification of images.

- consider output from current hierarchy level as input to next hierarchy level
- filter inputs in smaller sub-system for next hierarchy level
- keeping the same expected output and generating a fuzzy inference system or fuzzy unit for every sample set

Fig 4 represents the conceptual flow chart for hierarchical fuzzy system implementation [3] [4] [7]. At first step, collect training data, which consists of inputs as real image dataset and outputs as labels for the images. Use Otsu's multi-dimensional method to find multiple threshold values and multiply these values with each individual element of every input sample to generate a new input sample. In this process, the output values remain the same. However, the size of the matrix remains the same, only the values of the input element will be updated. Based on the samples of images, number of hierarchical levels required, number of fuzzy logic units per level and number of rules per fuzzy unit, the real data matrix is divided into number of segments of columns but with keeping the same output label. Every segment is defined with the same size and segment data is used to design hierarchical fuzzy logic units. Once the segmentation of data has been performed, the development of the first level of hierarchical level is made in such a way that the number of fuzzy logic units required for the first level is the same as the number of segments. In other words, every segment requires its own fuzzy logic unit. For every segment, generate a fuzzy inference system, evaluate the output of the unit and contain all the output from all the fuzzy logic units in a buffer. In aggregated hierarchical structure, all the real input data consumed at first level of the hierarchy and the next following hierarchical level consumes the output from the previous layer as input and maintains the same output as expected. In the final layer, the evaluated output will present the final output or label.

C. Synthesize using Xilinx ISE Environment

There has been an increasing interest about the hardware implementation of fuzzy systems. The FPGA environment has a significant presence in almost all the areas of research. For various researchers, it would have been a topic of future research to implement and synthesize the hierarchical fuzzy system using FPGA. The MATLAB environment has been used to develop the system and validate the given methodology in this paper. The simulation of any system in real time using an FPGA environment can be achieved by HDL languages [10]. To the best of authors' ability and knowledge, no researcher has conferred the steps to simulate and synthesize using HDL language especially Verilog HDL. Using embedded HDL Coder in MATLAB and Simulink, Verilog HDL code has been generated for the complete system. The main benefits of using Verilog HDL is its integrated capacity to create an overall system, generate code, simulate the environment and verify a system. The detailed steps below represent the procedure used in this research to automatically generate Verilog HDL using Simulink and design implementation using FPGA environment.

The first and foremost step is to auto-generate Verilog HDL code using Simulink embedded HDL coder. Once the Simulink model has been developed, go to model properties. In the code generation settings, choose code style parameter as: "Generate HDL option" and choose target code generation method as: "Verilog". Once the settings have been done, compile the Simulink model and click the generate button to auto-generate Verilog HDL code. Simulink embedded HDL coder creates link to Xilinx ISE environment and later executes the logic synthesis and 'place-and-route' steps in generated code.

The second step is the simulation and verification in an FPGA environment. ISim simulator has been used as the simulation environment for Verilog HDL code. Xilinx ISE has considered 'ARTIX-7 XC7A100T' as a target device with '4ns IO delay' for synthesizing the model. During synthesize the system, the design of desired circuit behavior transitioned to the design of logic gates implementation. The post synthesis generates the complete netlists. The command 'program' in Xilinx properties maps the logical design to FPGA hardware by performing DRC check on the implementation and mapping to target FPGA hardware device. In the next step, assign the cells of the FPGA device to a specific location, commonly called as cell placement. All the cells and resources of the FPGA hardware device are connected by the step called 'routing'. Once routing and placement is done, a bit stream has been created, which configures an FPGA device using Xilinx program iMPACT.

are shown in orange and blue color. The level of hierarchical system consists of fuzzy logic units designed for every segment and is connected in hierarchical architecture. For example, in case of aggregated hierarchical structure, all the fuzzy units designed using these segments must lie in lower level and output from lower level becomes the input for the next level and so on. On the other hand, for incremental hierarchical structure, every segment represents a single layer of hierarchical structure but apart from the first layer, output from the previous layer is also added as one of the inputs for the next layer. The above-described large dataset, i.e., YaleB, does not have any imprecise and uncertain information, as is real and prevalent for a large number of real-life problems. These databases have precise and fixed values contrary to imprecise and uncertain data.

V. ASSESSMENT OF RESULTS

The results of the hierarchical fuzzy system for image classification have been shown and discussed. The output of the hierarchical fuzzy system has been represented as the classification of images. The classification of image is the task of assigning a label to an input image from a set of predefined categories. The real-life image database such as YaleB database has been considered. The YaleB databases comprises large dataset of real-world images. Every image sample in the databases is assigned to the distinct label. In the YaleB database, the labels for the image dataset are classified as {0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ... 16}. In this paper, the output of the hierarchical fuzzy system is evaluated in the form of labels.

The main goal is to design a hierarchical fuzzy system-based algorithm for image classification that provides closer results to desired behavior i.e., expected labels. The accuracy of the classification of images is measured by comparing the labels of images defined in the database and the labels of images evaluated by the hierarchical fuzzy system. The labels of the hierarchical fuzzy system are compared with the labels evaluated using a conventional fuzzy system.

The following terminologies have been used as:

- **Expected output:** Labels defined in database for the classification of images
- **Actual output:** Evaluated Labels using fuzzy logic systems for the classification of images

In this section, following subsections have been presented and discussed in detail below:

- A. **Aggregation methods for hierarchical fuzzy logic unit:** This subsection discusses and specifies the selection criteria for the defuzzification aggregation methods for every hierarchical fuzzy logic unit.
- B. **Deviation and cumulative distribution:** This subsection describe the deviation of actual output with respect to expected labels and also presents the cumulative distribution function for detailed analysis
- C. **Accuracy of hierarchical systems:** This subsection shows the raw comparison of outputs (i.e., labels) has been made between labels from database and the labels evaluated by hierarchical fuzzy systems
- D. **Properties:** This subsection presents the additional behavioral properties of hierarchical systems. The comparison between labels from databases and the labels evaluated by hierarchical systems has been shown. Various parameters have been presented such as variance, standard deviation, median, mean, skewness, interquartile range and kurtosis.
- E. **Comparison of results with existing methodologies:** This subsection discusses the assessment of results in comparison with existing methodologies such as neural networks. The accuracy and computational time have been compared between the hierarchical approach presented in this paper and the existing methodologies such as CNNs.

A. Aggregation methods for hierarchical fuzzy logic unit

The analysis for the selection criteria of aggregation methods for designing a low dimension hierarchical fuzzy system is presented by Radek Sindelar [6]. In this paper, the analysis of two weighted approaches [6] i.e., maximum (max) aggregation method and weighted average aggregation method has been compared and observed for hierarchical fuzzy systems.

Hierarchical fuzzy system comprises several fuzzy logic units at various levels of hierarchy. Each fuzzy logic unit has its own rule base extracted from clusters of data. In hierarchical structure, output of previous level becomes the input to the next level. Hence, it is very important to select the effective aggregation method in defuzzification for every fuzzy logic unit. The selection of aggregation method provides better and closed results compared to desired output i.e., expected labels. The graphical representation and comparison of the both the aggregation methods in reference to conventional fuzzy are shown in Fig 7. Where 'sFIS' represents the conventional system and 'hFIS' represents the hierarchical fuzzy system.

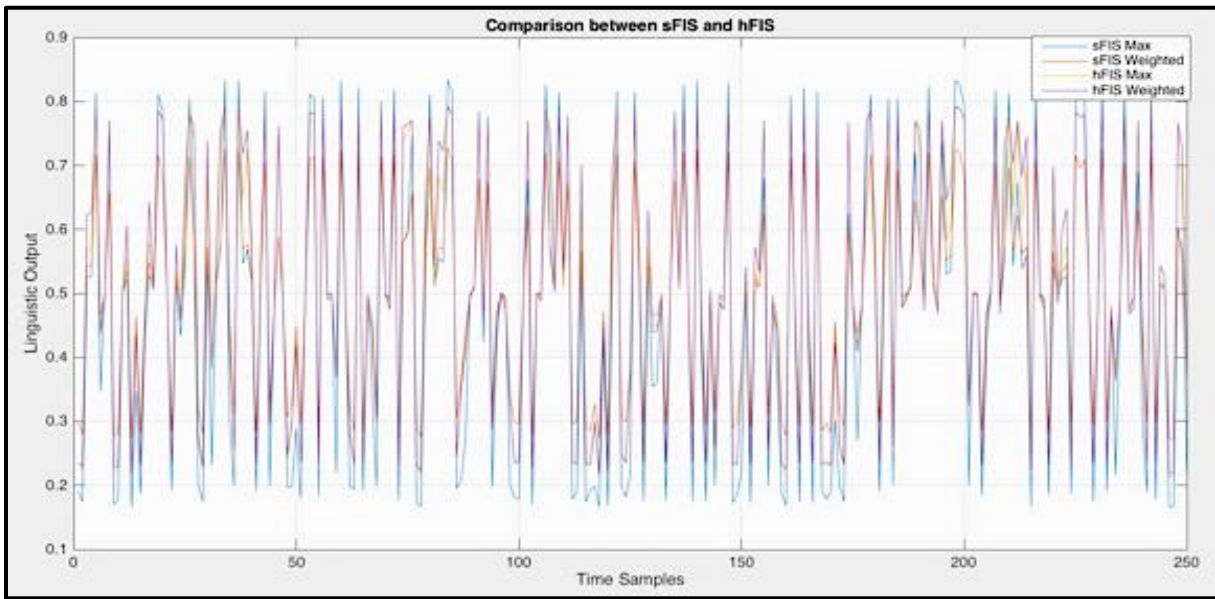


Fig 7. Comparison of aggregation methods

It is observed that the weighted average as an aggregation method provides the better and closer result compared to the maximum aggregation method. It is noted that while analyzing both situations, the same model and the rule base have been considered. The correlation for two aggregation methods between conventional and hierarchical fuzzy systems are defined below. Here, the correlation value describes the relative variation of actual output in compared to expected output. Higher the value of correlation value gives low deviation and thus provides more accuracy. It is observed that the correlation between sFIS and hFIS for Maximum aggregation method is 0.84 and for Weighted average aggregation method: 0.92.

B. Deviation and cumulative distribution

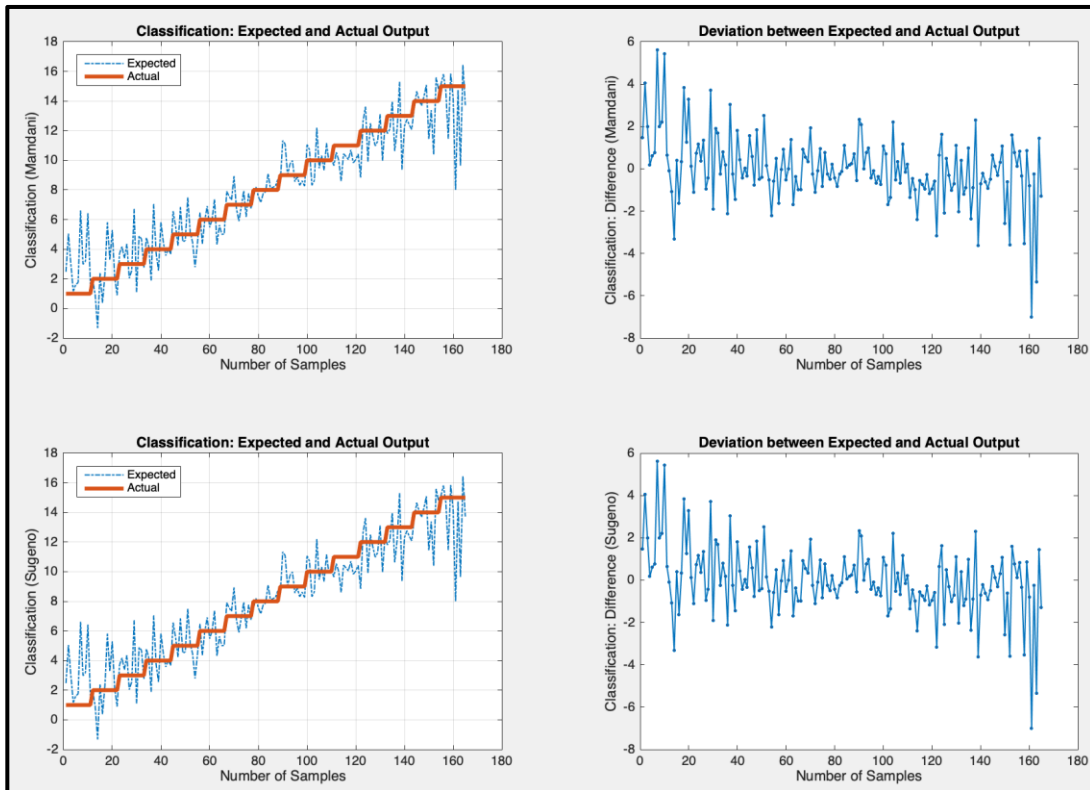


Fig 8. Deviation between actual and expected result

The algorithm designs a system with higher interpretability and accuracy i.e., close to expected labels as possible. The comparison between expected output and actual output has been shared in the following figures. Fig 8 represents the classification of images for expected and actual output. It also presents the deviation of outputs.

The deviation is calculated by the relative difference between expected output and actual output. It can be seen that the output deviation between expected and actual output lies close to zero. More the deviation closer to zero, higher the accuracy of the systems will be. The descriptions of four graphs are given as:

- **Top-left plot:** Mamdani approach – Actual and expected output behavior
- **Top-right plot:** Mamdani approach – Deviation of actual output compared to expected output
- **Bottom-left plot:** Sugeno approach – Actual and expected output behavior
- **Bottom-right plot:** Sugeno approach – Deviation of actual output compared to expected output

Where, x-axis presents the number of image samples and y-axis represents the classification of images i.e., image labels for the data. It has observed that deviation of output values (i.e., labels) between labels for images defined in the database and labels for images evaluated by hierarchical fuzzy systems for both Mamdani and Sugeno type fuzzy inference systems are identical.

Fig 9 shows the empirical cumulative distribution function (eCDF) between expected and actual output. The empirical cumulative distribution function plots the output data in the order of least to greatest absolute value and shows the distribution across the dataset.

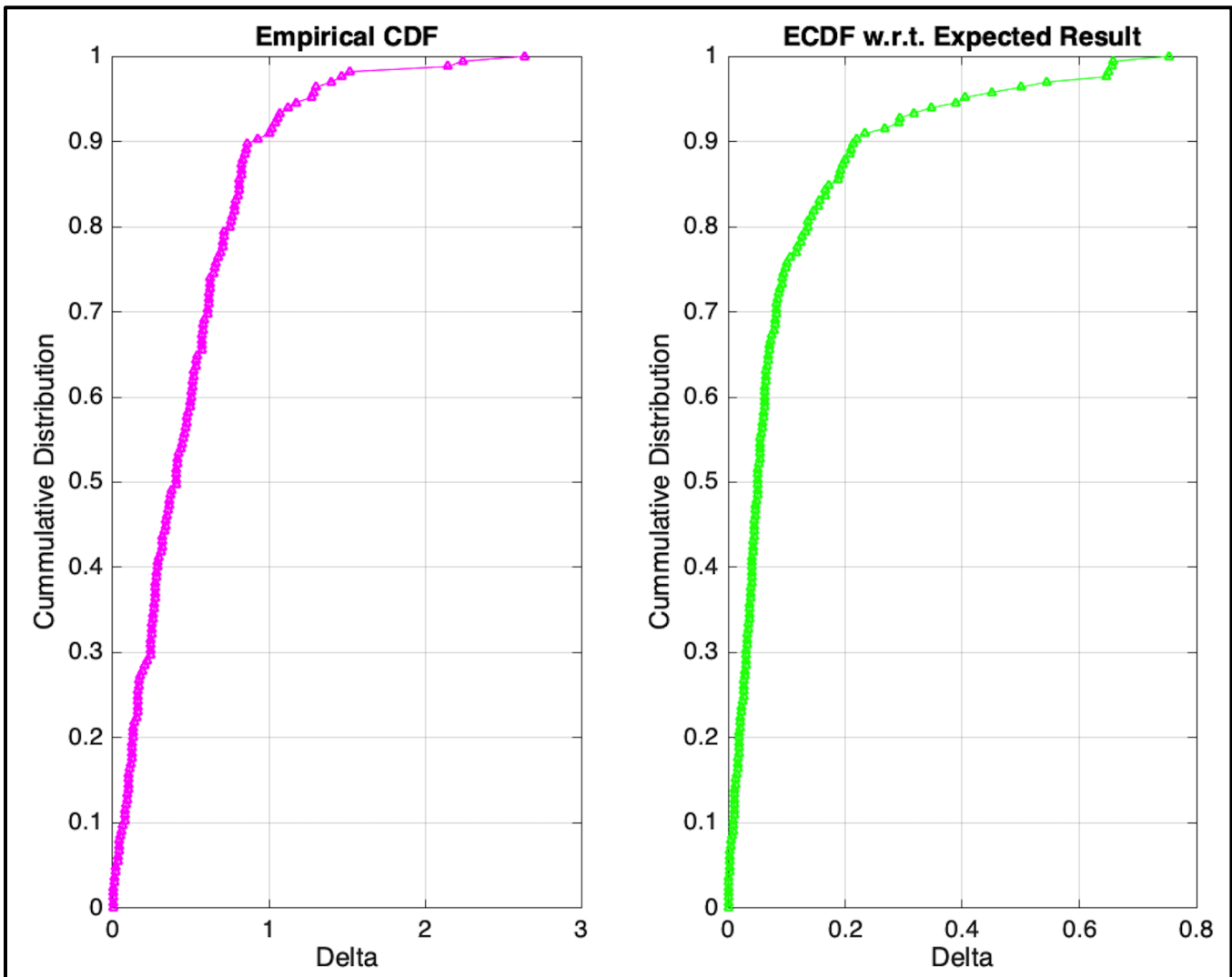


Fig 9. Cumulative distribution function between expected versus actual results

In this paper, eCDF function has been used to detect the accuracy and deviation of the overall system. This can be achieved by analyzing the distribution of data within 5% of error tolerance, 10% of error tolerance or higher. More data lies inside the 5% or 10% of error tolerance, higher the accuracy of the overall system. In other words, if the system is less prone to error, classification of images will be accurate. The graph on the left hand side represents eCDF in reference to labels evaluated by hierarchical fuzzy systems, whereas the graph on right hand side represents eCDF in reference to labels defined in the database.

The consecutive difference between the output label values of YaleB database is approximately one. The figure on the left-hand side shows that the 90% of the data falls within the range of delta less than 1. The figure on the right-hand side presents the absolute difference between expected and actual output. It is clearly visible in the graph (right hand side) and observed in eCDF that more than 80% of labels evaluated by hierarchical fuzzy systems are similar to and within tolerance of 10% from the labels for images defined in the database.

C. Accuracy of hierarchical System for image classification

Fig 10 shows the comparison between expected output versus raw actual output for complete samples in ascending order of labels. It is visible that the behavior between expected and actual is similar. This showcases the efficiency of the hierarchical fuzzy systems. Keep in mind, the system has been designed with the aggregation method as the weighted average.

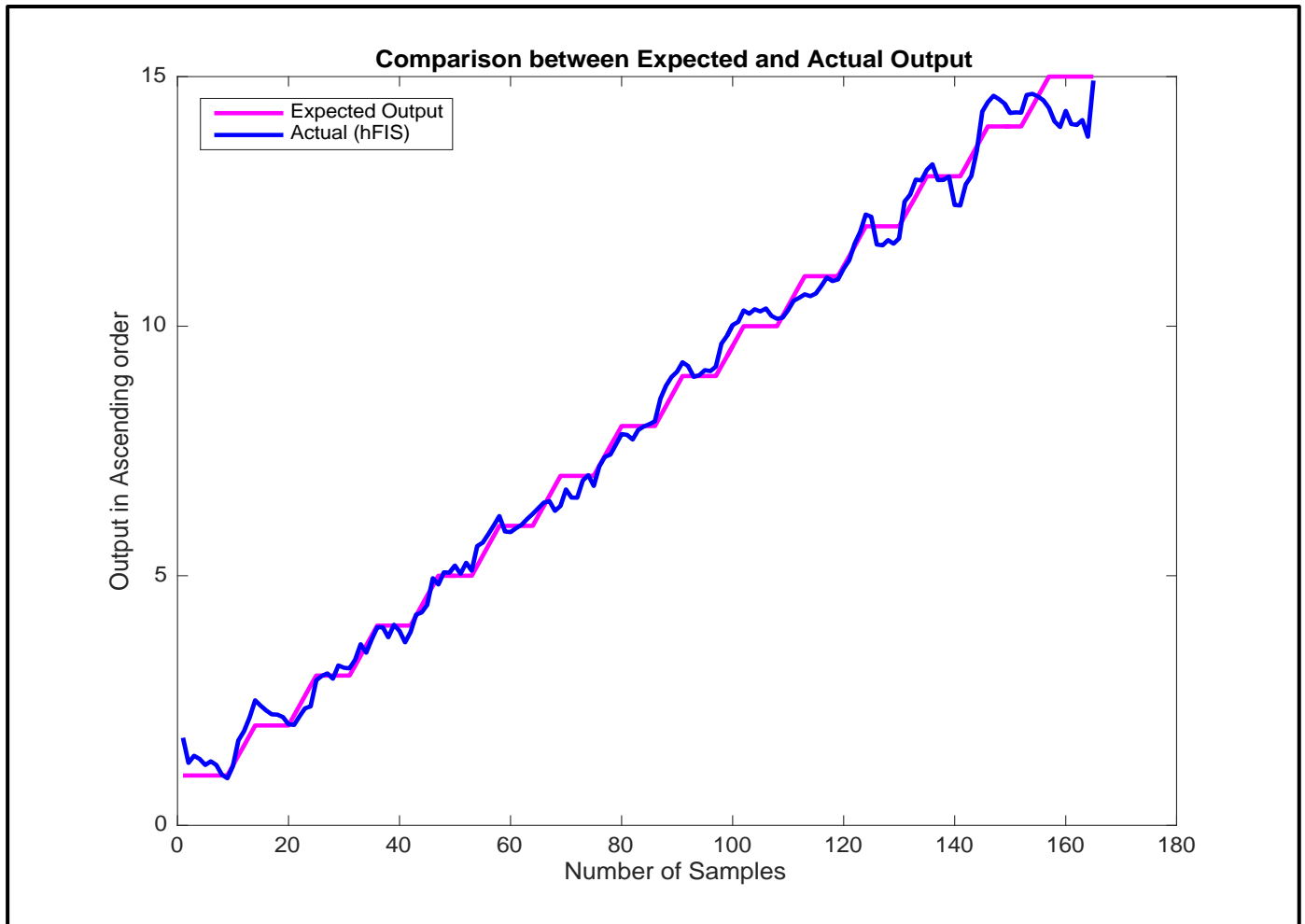


Fig 10. Accuracy of image classification between expected and actual output (labels)

Where, x-axis shows the number of samples and y-axis shows the output(s) as expected and actual output. The output(s) have been arranged in ascending order to showcase the behavior of two outputs adjacent to each other.

D. Properties

TABLE 1 represents the behavioral properties of expected and actual output with following parameters in the MATLAB environment. It is noted that the same model and same rule base have been considered for analyzing these properties.

TABLE 1
RAW OUTPUT COMPARISON BETWEEN EXPECTED AND ACTUAL OUTPUT

| Parameter(s) | Description of Parameter(s) | Expected Output (labels in dataset) | Actual Output (evaluated labels) |
|----------------------------|---|-------------------------------------|----------------------------------|
| Variance | Computes the consistency or dispersion of data over specified vector dimension | 18.78 | 18.37 |
| Standard Deviation | Measure amount of variation or dispersion within data set or specified dimension | 4.33 | 4.33 |
| Median | Value separating data sample in two halves: higher and lower; or middle value | 8 | 8 |
| Mean | Average of all the values in dataset | 8 | 8 |
| Skewness | Measure of asymmetry of the probability distribution of real data around its mean value | 0 | 0 |
| Interquartile Range | Measure of statistical dispersion equally distributed between 25 th and 75 th percentile. It is the measure of variability depending on the division of data in quartiles | 8 | 8 |
| Kurtosis | Peak value of a frequency distribution | 1.79 | 1.79 |

E. Comparison of results with existing methodology

There are numerous deep learning methods such as the convolution neural networks (CNN) open-source libraries [15] exist in the literature. For CNN models, MATLAB has a toolbox for computer vision applications called MatConvNet [15] [16]. These CNNs are composed of linear and non-linear filtration operations such as convolution and rectification. The benefit of using MatConvNet is its ability to provide an efficient environment for researchers. In this paper, MatConvNet toolbox has been used to learn YaleB database using CNN model for comparison. Table 2 displays the accuracy comparison between the results from CNN model trained via MatConvNet and hierarchical fuzzy system presented in this paper. It is noted that YaleB database has been considered for both the approaches. As shown in Table 2, the outputs spreading is calculated as the difference between expected output and actual observed output.

TABLE 2
ACCURACY COMPARISON BETWEEN CNN MODEL AND HIERARCHICAL FUZZY MODEL

| Models and Methodologies | Outputs spread within 10% error tolerance | Outputs spread in more than 10% error tolerance |
|--|---|---|
| Outputs from CNN Model computed via MatConvNet Toolbox | 91.2% | 8.8% |
| Outputs from Hierarchical Fuzzy System using Image Thresholding | 90% | 10% |

Table 3 shows the computational time comparison between convolution neural network model and hierarchical fuzzy system. The computational time represents the time taken to generate a complete model from the YaleB database. The computational time can be reduced further by calibrating the number of clusters required. Here, the number of clusters represent the total rules required by each fuzzy logic unit of hierarchical structure. For better performance, it is necessary to calibrate the system to the minimum number of clusters where desired accuracy can be met.

TABLE 3
COMPUTATION TIME BETWEEN CNN MODEL USING MATCONVNET AND HIERARCHICAL FUZZY MODEL

| Models and Methodologies | Description | Computation Time in Seconds |
|----------------------------------|---|-----------------------------|
| CNN Model via MatConvNet Toolbox | Number of Epochs: 20 | 25.6 |
| Hierarchical Fuzzy System | Number of Clusters per fuzzy logic unit: 20 | 15.1 |
| | Number of Clusters per fuzzy logic unit: 10 | 11.7 |

It is observed by comparing the accuracy results obtained between two methodologies that the convolution neural networks have little edge over hierarchical fuzzy system approach. However, the computation time of hierarchical approach is significantly better than CNN model. One of the major benefits of hierarchical fuzzy systems is its ability to further reduce the computation time by calibrating the number of required clusters for every fuzzy logic unit. In a large number of real-life applications such as defense and security, the computational time is extremely important, and the hierarchical fuzzy system provides better and reduced computation time compared to existing convolution neural networks models. It is noted that the YaleB database contains finite and precise image datasets. It is expected that with more uncertain, imprecise and vague nature of data such as real-life human assessments, the hierarchical approach will overcome existing models in accuracy too.

VI. CONCLUSION

In this paper, a brief review of the deep learning approaches for conventional systems such as fuzzy logic and neuro-fuzzy systems is given first. Most of the complex real-life data including human assessments are uncertain, imprecise and vague in nature. To overcome the limitations of conventional systems, the design of the new generalized algorithm for large intelligent systems using hierarchical fuzzy systems is proposed. The hierarchical system leads to a smaller number of rule-bases and has reduced overall complexity of the system compared to a conventional fuzzy system. The algorithm presented in this paper has the capacity to handle uncertain and imprecise datasets. The new algorithm is developed with the help of Otsu's multi-dimensional approach and is implemented using real image databases such as YaleB database for face recognition etc. The results for the hierarchical fuzzy systems for the large image data have been presented. The accuracy and computational time have been compared between the hierarchical approach presented in this paper and existing methodologies such as convolution neural networks. It is clearly shown that computational time of hierarchical approach is much better than existing methodologies in the literature. This algorithm can be extended to any hierarchical tree structure. The generalized algorithm can be applied for both Type-1 and Type-2 fuzzy systems. The future work involves the implementation of real life human-perception dataset i.e., impreciseness and uncertainty. It is hoped that results obtained in this paper will be helpful for handling a variety of advanced image processing real-life applications.

REFERENCES

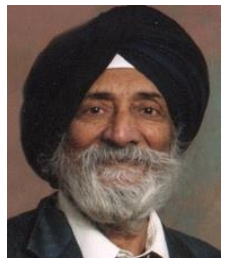
- [1] S. Kamthan and H. Singh, "Hierarchical Fuzzy Logic for Multi-Input Multi-Output Systems," in *IEEE Access* 2020, doi: 10.1109/ACCESS.2020.3037901, 2020
- [2] R. Scherer, L. Rutkowski, "A survey of hierarchical fuzzy systems", Proceedings of the fifth conference neural networks and soft computing, Zakopane, pp. 6-10, 2000
- [3] S. Kamthan, H. Singh, and T. Meitzler, "UAVs: on development of fuzzy model for categorization of countermeasures during threat assessment." *SPIE Defense+ Security*, pp. 1019518-1019518, International Society for Optics and Photonics, 2017
- [4] S. Kamthan, H. Singh and T. Meitzler, "Survivability: a hierarchical fuzzy logic layered model for threat management of unmanned ground vehicles." *Defense + Security*, 2018
- [5] H. Singh, M. M. Gupta, T. Meitzler, Z. G. Hou, K. K. Garg, A. M. Solo, and L. A. Zadeh, "Real-Life applications of fuzzy logic." *Adv. fuzzy systems* 2013: 581879-1. APA, 2013
- [6] R. Sindelar, "Hierarchical Fuzzy Systems", *IFAC Proceedings Volumes* 38, no. 1, pp 245-250, 2005
- [7] L. Sun Liang, H. Wei, "Adaptive fuzzy control of spacecraft proximity operations using hierarchical fuzzy systems." *IEEE/ASME transactions on mechatronics* 21, no. 3: 1629-1640, 2016
- [8] P. W. Simões, B. I. Narjara, S. C. Ramon, V. Ramon, D. V. Carlos, P. M. Gustavo, L. Edroaldo et al. "Classification of images acquired with colposcopy using artificial neural networks." *Cancer informatics* 13, CIN-S17948, 2014

- [9] Y. Jarraya, S. Bouaziz, H. Hagra, and A. M. Alimi, "A Multi-Agent Architecture for the Design of Hierarchical Interval Type-2 Beta Fuzzy System." IEEE Transactions on Fuzzy Systems 27, no. 6, pp 1174-1188, 2018
- [10] T. Yamakawa, "Hardware Implementation of Fuzzy Logic Controller -Designability, Stability and Robustness", in Control and Systems Engineering, Studies in Systems, Decision and Control, vol 27., pp. 261-273, Springer, Cham 2015
- [11] J. Gu, H. Hu, and H. Li, "Local robust sparse representation for face recognition with single sample per person", IEEE/CAA Journal of Automatica Sinica, sn 5(2), pp 547-554, 2017
- [12] L. A. Zadeh, "Fuzzy sets." Information and control 8.3 (1965): 338-353
- [13] Y. Lin and C. Songcan, "A Centroid auto-fused hierarchical fuzzy c-means clustering" IEEE Transactions on Fuzzy Systems, pp 99, sn 1941-0034, April 2020
- [14] R. M. Asl, R. Palm, H. Wu, and H. Handroos, "Fuzzy-based parameter optimization of adaptive unscented Kalman filter: Methodology and experimental validation." IEEE Access 8, 2020, pp 54887-54904.
- [15] A. Vedaldi, K. Lenc, "Matconvnet: Convolution neural networks for matlab", Proceedings of the 23rd ACM international conference on Multimedia, pp. 689-692, 2015
- [16] A. Vedaldi, M. Lux and M. Bertini, "MatConvNet: CNNs are also for MATLAB users", ACM SIG Multimedia Records 10, no. 1, pp 9-9, 2018
- [17] L. Sun and W. Huo, "Adaptive fuzzy control of spacecraft proximity operations using hierarchical fuzzy systems," IEEE/ASME Trans. Mechatronics, vol. 21, no. 3, pp. 1629-1640, Jun. 2016



Shashank Kamthan (M'2014) received Bachelor of Engineering in 2007 from Rajiv Gandhi Proudyogiki Vishwavidhyalaya, India and Master of Science in Electrical Engineering in the year 2008 from Wayne State University, Detroit, MI, USA. He is currently pursuing Ph.D. from Wayne State University, MI, USA. He has more than 12 publications in international journals and conferences. On the professional front, he has over 10 years of professional experience. During his stint with the automotive industry, he has worked in the field of model-based development, AUTOSAR, connected vehicle systems (V2X), embedded software development, systems integration etc. At present, he is working full-time as Senior Embedded Systems Consultant in Automotive Domain. His current areas of interest are target detection, artificial intelligence, connected vehicles, deep learning techniques, autonomous vehicles, mobility,

controls, model-based design etc.



Harpreet Singh (M'73, SM'85, LSM'2009) received his B.Sc. Eng. degree from Punjabi University, India, in 1963 and his M.S. and Ph.D. degrees from the University of Roorkee, India, in 1966 and 1971. He taught at the University of Roorkee from 1963 to 1981. Since 1981 he has been with Wayne State University, Detroit, MI, where he has been a professor in the Department of Electrical and Computer Engineering.

He has worked in diversified areas of systems, networks, controls, fuzzy logic, DNA Nano-digital circuits, bio-circuits and VLSI design. Dr. Singh has more than 300 publications in international journals and conferences and has received a number of awards. He has organized a number of national and international conferences and has supervised more than 25 Ph.D. theses in different areas of electrical and computer engineering. He has been awarded the distinguished Alumni Award of prestigious IIT Roorkee in 2012. He has also been awarded the distinguished alumni

award from President of India in the year 1983 from Thapar Engineering College, Patiala. As a founder chair of Life Affinity Group of South Eastern Michigan section, the section received 2017 IEEE Life Member Affinity Group Achievement award.



Thomas J. Meitzler received his B.S. and M.S. in Physics from Eastern Michigan University, completed graduate coursework at the University of Michigan, and received a Ph.D. in Electrical Engineering from Wayne State University in Detroit. His doctoral dissertation in Electrical Engineering at Wayne State University was on Modern Methods for Computing the Probability of Target Detection in Cluttered Environments. He is a Fellow of the American Physical Society (APS) and Senior Member of the Institute for Electrical and Electronics Engineers (IEEE) and Society of Photo-optical Instrumentation Engineers (SPIE). He is the Ground System Survivability Senior Science Advisor. During the time from 1988 to present, Dr.

Meitzler has been a research engineer at the US Army TACOM Research and Engineering Center (TARDEC) in Survivability. For the U.S. Army, Dr. Meitzler has been involved with the validation, verification, and development of infrared, electro-optical and human visual acquisition models. Dr. Meitzler was the principal scientist of the TARDEC Visual Perception Laboratory and the principal investigator on Cooperative Research and Development Agreements (CRADA's) with GM and Ford Motor Company to apply visual target acquisition models to vehicle conspicuity and novel sensors to automobile 360-degree safety. Dr. Meitzler was the principal investigator on several Space Act Agreements with NASA's Kennedy Space Center and CRADA's with the Columbia University College of Physicians and Surgeons. He has authored/co-authored many papers in the area of Electro Optic system simulation, visual detection, sensor validation, traditional nondestructive armor evaluation and embedded piezoelectric nondestructive testing of armor materials, and spintronics. Dr. Meitzler has developed and integrated technologies for embedded armor health-monitoring, armor Non-Destructive testing, and armor embedded radio signal detection.